Engi's

Generated by Doxygen 1.8.15

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

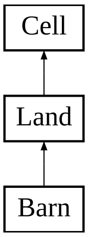
Chapter 4

Class Documentation

4.1 Barn Class Reference

```
#include <Barn.h>
```

Inheritance diagram for Barn:



Public Member Functions

- **Category getCategory** () const

Static Private Attributes

- static constexpr **Category category** { **BARN**}

4.1.1 Member Function Documentation

4.1.1.1 getCategory()

```
Category Barn::getCategory ( ) const [virtual]
```

Return kategori dari objek ini

Implements **Cell** (p. 13).

4.1.2 Member Data Documentation

4.1.2.1 category

```
constexpr Category Barn::category { BARN} [static], [private]
```

Menandakan bahwa land bertipe **Barn** (p. 7)

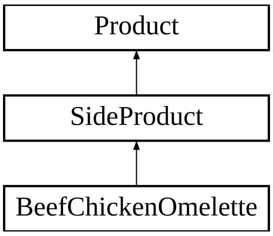
The documentation for this class was generated from the following file:

- **Barn.h**

4.2 BeefChickenOmelette Class Reference

```
#include <BeefChickenOmelette.h>
```

Inheritance diagram for BeefChickenOmelette:



Public Member Functions

- **BeefChickenOmelette** ()
- int **getPrice** () const
- **Category getCategory** () const

Static Public Member Functions

- static **LinkedList**< **Product** ∗ > & **getRecipe** ()

Static Private Attributes

- static constexpr int **price** {250000}
- static constexpr **Category category** { **BEEFCHICKENOMELETTE**}
- static **LinkedList**< **Product** ∗ > **recipe**

4.2.1 Constructor & Destructor Documentation

4.2.1.1 BeefChickenOmelette()

```
BeefChickenOmelette::BeefChickenOmelette ( )
```

Constructor untuk inisialisasi recipe

4.2.2 Member Function Documentation

4.2.2.1 getCategory()

```
Category BeefChickenOmelette::getCategory ( ) const [virtual]
```

Mengembalikan category dari produk

Implements **Product** (p. 66).

4.2.2.2 getPrice()

```
int BeefChickenOmelette::getPrice ( ) const [virtual]
```

getPrice mengembalikan harga yang didefinisikan

Implements **Product** (p. 66).

4.2.2.3 getRecipe()

```
static LinkedList< Product*>& BeefChickenOmelette::getRecipe ( ) [static]
```

Mengembalikan resep dari produk

4.2.3 Member Data Documentation

4.2.3.1 category

```
constexpr Category BeefChickenOmelette::category { BEEFCHICKENOMELETTE} [static], [private]
```

Kategori dari **BeefChickenOmelette** (p. 8)

4.2.3.2 price

```
constexpr int BeefChickenOmelette::price {250000}  [static], [private]
```

Harga dari **BeefChickenOmelette** (p. 8)

4.2.3.3 recipe

```
LinkedList< Product*> BeefChickenOmelette::recipe  [static], [private]
```

Resep **BeefChickenOmelette** (p. 8). Terdiri dari **CowMeat** (p. 23) dan **ChickenEgg** (p. 17).

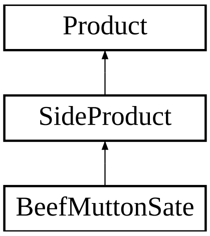
The documentation for this class was generated from the following file:

- **BeefChickenOmelette.h**

4.3 BeefMuttonSate Class Reference

```
#include <BeefMuttonSate.h>
```

Inheritance diagram for BeefMuttonSate:



Public Member Functions

- **BeefMuttonSate** ()
- int **getPrice** () const
- **Category getCategory** () const

Static Public Member Functions

- static **LinkedList**< **Product** ∗ > & **getRecipe** ()

Static Private Attributes

- static const int **price** {404000}
- static constexpr **Category category** { **BEEFMUTONSATE**}
- static **LinkedList**< **Product** ∗ > **recipe**

4.3.1 Constructor & Destructor Documentation

4.3.1.1 BeefMuttonSate()

```
BeefMuttonSate::BeefMuttonSate ( )
```

Constructor untuk inisialisasi recipe

4.3.2 Member Function Documentation

4.3.2.1 getCategory()

```
Category BeefMuttonSate::getCategory ( ) const  [virtual]
```

Mengembalikan category dari produk

Implements **Product** (p. 66).

4.3.2.2 getPrice()

```
int BeefMuttonSate::getPrice ( ) const  [virtual]
```

getPrice mengembalikan harga yang didefinisikan

Implements **Product** (p. 66).

4.3.2.3 getRecipe()

```
static LinkedList< Product*>& BeefMuttonSate::getRecipe ( )  [static]
```

Mengembalikan resep dari produk

4.3.3 Member Data Documentation

4.3.3.1 category

```
constexpr  Category BeefMuttonSate::category { BEEFMUTONSATE}  [static], [private]
```

Kategori dari **BeefMuttonSate** (p. 10)

4.3.3.2 price

```
const int BeefMuttonSate::price {404000}  [static], [private]
```

Harga dari **BeefMuttonSate** (p. 10)

4.3.3.3 recipe

```
LinkedList< Product*> BeefMuttonSate::recipe  [static], [private]
```

Resep **BeefMuttonSate** (p. 10). Terdiri dari **CowMeat** (p. 23) dan **SheepMeat** (p. 68).

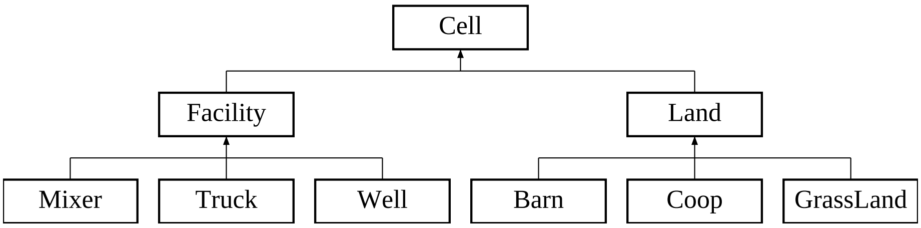
The documentation for this class was generated from the following file:

- **BeefMuttonSate.h**

4.4 Cell Class Reference

```
#include <Cell.h>
```

Inheritance diagram for Cell:



Public Member Functions

- virtual ∼**Cell** ()=0
- virtual bool **isFacility** () const =0
- virtual **Category getCategory** () const =0
- bool **getIsOcupied** ()
- void **setIsOcupied** (bool)
- virtual void **growGrass** ()
- virtual void **removeGrass** ()
- virtual bool **isGrassExist** () const =0

Private Attributes

- bool **isOcupied** {false}

4.4.1 Constructor & Destructor Documentation

4.4.1.1 ∼Cell()

```
virtual Cell::∼Cell ( )  [pure virtual]
```

dtor untuk **Cell** (p. 12)

4.4.2 Member Function Documentation

4.4.2.1 getCategory()

```
virtual  Category Cell::getCategory ( ) const  [pure virtual]
```

Return categori dari objek kategori

Implemented in **Barn** (p. 7), **Coop** (p. 20), **GrassLand** (p. 37), **Mixer** (p. 57), **Truck** (p. 72), and **Well** (p. 73).

4.4.2.2 getIsOcupied()

```
bool Cell::getIsOcupied ( )
```

Mengambil nilai boolean isOcupied

4.4.2.3 growGrass()

```
virtual void Cell::growGrass ( )  [virtual]
```

Menambah air pada cell. Jika bertipe **Land** (p. 41) akan menumbuhkan rumput. Jika tidak, tidak akan berefek apa-apa.

Reimplemented in **Land** (p. 42).

4.4.2.4 isFacility()

```
virtual bool Cell::isFacility ( ) const  [pure virtual]
```

Return true jika objek adalah **Facility** (p. 31)

Implemented in **Facility** (p. 32), and **Land** (p. 42).

4.4.2.5 isGrassExist()

```
virtual bool Cell::isGrassExist ( ) const  [pure virtual]
```

Mengembalikan keberadaan grass jika **Cell** (p. 12) bertipe **Land** (p. 41)

Implemented in **Land** (p. 42), and **Facility** (p. 32).

4.4.2.6 removeGrass()

```
virtual void Cell::removeGrass ( )  [virtual]
```

Reimplemented in **Land** (p. 42).

4.4.2.7 setIsOcupied()

```
void Cell::setIsOcupied (
             bool  )
```

Mengganti nilai boolean isOcupied

4.4.3 Member Data Documentation

4.4.3.1 isOcupied

```
bool Cell::isOcupied {false}  [private]
```

Flag yang menandakan cell ditempati oleh sesuatu (Player/FarmAnimal/Facility) atau tidak. True bila cell sedang ditempati oleh sesuatu.

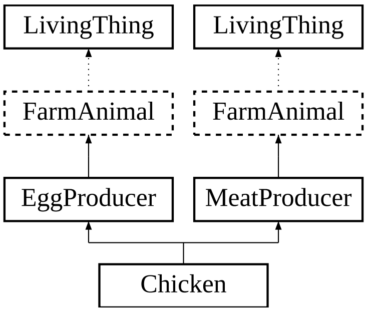
The documentation for this class was generated from the following file:

- **Cell.h**

4.5 Chicken Class Reference

`#include <Chicken.h>`

Inheritance diagram for Chicken:



Public Member Functions

- **Chicken** (**Point position**, **Cell** ∗∗∗& **worldMap**, int **nRowCell**, int **nCollumnCell**)
- **FarmProduct** ∗ **ProduceProduct** (**Action**) const
- std::string **makeNoise** () const

Private Member Functions

- virtual bool **canMoveTo** (**Cell** toWhere) const

Static Private Attributes

- static constexpr int **maxTimeToGetHungryChicken** {15}

4.5.1 Constructor & Destructor Documentation

4.5.1.1 Chicken()

```
Chicken::Chicken (
              Point position,
              Cell ***& worldMap,
             int nRowCell,
             int nCollumnCell )
```

Constructor

4.5.2 Member Function Documentation

4.5.2.1 canMoveTo()

```
virtual bool Chicken::canMoveTo (
              Cell toWhere ) const  [private], [virtual]
```

Mengecek apakah bisa pindah (tidak out of bound, bertipe **Coop** (p. 20) atau **GrassLand** (p. 37), tidak ada hewan lain)

Reimplemented from **EggProducer** (p. 30).

4.5.2.2 makeNoise()

```
std::string Chicken::makeNoise ( ) const  [virtual]
```

Mengembalikan suara dari **Chicken** (p. 15)

Implements **FarmAnimal** (p. 34).

4.5.2.3 ProduceProduct()

```
 FarmProduct* Chicken::ProduceProduct (
              Action  ) const
```

Mengembalikan FarmProduk yang akan dihasilkan **Chicken** (p. 15) bila **Chicken** (p. 15) di kill

4.5.3 Member Data Documentation

4.5.3.1 maxTimeToGetHungryChicken

```
constexpr int Chicken::maxTimeToGetHungryChicken {15}  [static], [private]
```

Nilai dari maxTimeToGetHungry

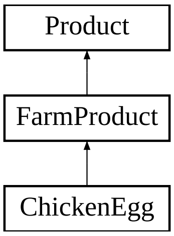
The documentation for this class was generated from the following file:

- **Chicken.h**

4.6 ChickenEgg Class Reference

```
#include <ChickenEgg.h>
```

Inheritance diagram for ChickenEgg:



Public Member Functions

- int **getPrice** () const
- **Category** **getCategory** () const

Static Private Attributes

- static const int **price** {2000}
- static constexpr **Category** **category** { **CHICKENEGG**}

4.6.1 Member Function Documentation

4.6.1.1 getCategory()

```
Category ChickenEgg::getCategory ( ) const [virtual]
```

Mengembalikan category dari produk

Implements **Product** (p. 66).

4.6.1.2 getPrice()

```
int ChickenEgg::getPrice ( ) const [virtual]
```

getPrice mengembalikan harga yang didefinisikan

Implements **Product** (p. 66).

4.6.2 Member Data Documentation

4.6.2.1 category

```
constexpr  Category ChickenEgg::category { CHICKENEGG}  [static], [private]
```

Kategori **ChickenEgg** (p. 17)

4.6.2.2 price

```
const int ChickenEgg::price {2000}  [static], [private]
```

Harga dari **ChickenEgg** (p. 17)

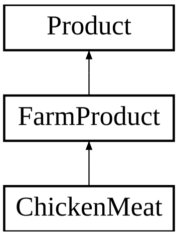
The documentation for this class was generated from the following file:

- **ChickenEgg.h**

4.7 ChickenMeat Class Reference

```
#include <ChickenMeat.h>
```

Inheritance diagram for ChickenMeat:



Public Member Functions

- int **getPrice** () const
- **Category getCategory** () const

Static Private Attributes

- static const int **price** {20000}
- static constexpr **Category category** { **CHICKENMEAT**}

4.7.1 Member Function Documentation

4.7.1.1 getCategory()

```
 Category ChickenMeat::getCategory ( ) const  [virtual]
```

Mengembalikan category dari produk

Implements **Product** (p. 66).

4.7.1.2 getPrice()

```
int ChickenMeat::getPrice ( ) const  [virtual]
```

getPrice mengembalikan harga yang didefinisikan

Implements **Product** (p. 66).

4.7.2 Member Data Documentation

4.7.2.1 category

```
constexpr  Category ChickenMeat::category { CHICKENMEAT}  [static], [private]
```

Kategori dari **ChickenMeat** (p. 18)

4.7.2.2 price

```
const int ChickenMeat::price {20000}  [static], [private]
```

Harga dari **ChickenMeat** (p. 18)

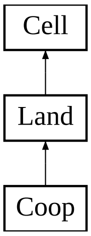
The documentation for this class was generated from the following file:

- **ChickenMeat.h**

4.8 Coop Class Reference

```
#include <Coop.h>
```

Inheritance diagram for Coop:



Public Member Functions

- **Category getCategory** () const

Static Private Attributes

- static constexpr **Category category** { **COOP**}

4.8.1 Member Function Documentation

4.8.1.1 getCategory()

```
Category Coop::getCategory ( ) const [virtual]
```

Return kategori dari objek ini

Implements **Cell** (p. 13).

4.8.2 Member Data Documentation

4.8.2.1 category

```
constexpr Category Coop::category { COOP} [static], [private]
```

Menandakan bahwa land bertipe **Coop** (p. 20)

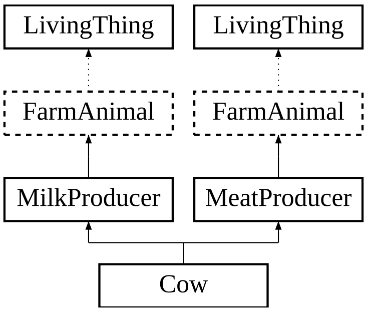
The documentation for this class was generated from the following file:

- **Coop.h**

4.9 Cow Class Reference

```
#include <Cow.h>
```

Inheritance diagram for Cow:



Public Member Functions

- **Cow** (**Point position**, **Cell** ∗∗∗& **worldMap**, int **nRowCell**, int **nCollumnCell**)
- **FarmProduct** ∗ **ProduceProduct** (**Action**) const
- std::string **makeNoise** () const

Private Member Functions

- virtual bool **canMoveTo** (**Cell** toWhere) const

Static Private Attributes

- static constexpr int **maxTimeToGetHungryCow** {20}

4.9.1 Constructor & Destructor Documentation

4.9.1.1 Cow()

```
Cow::Cow (
              Point position,
              Cell ***& worldMap,
             int nRowCell,
             int nCollumnCell )
```

Constructor

4.9.2 Member Function Documentation

4.9.2.1 canMoveTo()

```
virtual bool Cow::canMoveTo (
              Cell toWhere ) const  [private], [virtual]
```

Mengecek apakah bisa pindah (tidak out of bound, bertipe **Barn** (p. 7) atau **GrassLand** (p. 37), tidak ada hewan lain)

Reimplemented from **MeatProducer** (p. 55).

4.9.2.2 makeNoise()

```
std::string Cow::makeNoise ( ) const  [virtual]
```

Mengembalikan suara dari **Cow** (p. 21)

Implements **FarmAnimal** (p. 34).

4.9.2.3 ProduceProduct()

```
 FarmProduct* Cow::ProduceProduct (
              Action  ) const
```

Mengembalikan FarmProduk yang akan dihasilkan **Cow** (p. 21) bila **Cow** (p. 21) di kill

4.9.3 Member Data Documentation

4.9.3.1 maxTimeToGetHungryCow

```
constexpr int Cow::maxTimeToGetHungryCow {20}  [static], [private]
```

Nilai dari maxTimeToGetHungry

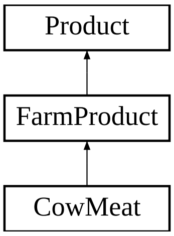
The documentation for this class was generated from the following file:

- **Cow.h**

4.10 CowMeat Class Reference

```
#include <CowMeat.h>
```

Inheritance diagram for CowMeat:



Public Member Functions

- int **getPrice** () const
- **Category** **getCategory** () const

Static Private Attributes

- static const int **price** {200000}
- static constexpr **Category** **category** { **COWMEAT**}

4.10.1 Member Function Documentation

4.10.1.1 getCategory()

```
Category CowMeat::getCategory ( ) const  [virtual]
```

Mengembalikan category dari produk

Implements **Product** (p. 66).

4.10.1.2 getPrice()

```
int CowMeat::getPrice ( ) const  [virtual]
```

getPrice mengembalikan harga yang didefinisikan

Implements **Product** (p. 66).

4.10.2 Member Data Documentation

4.10.2.1 category

```
constexpr Category CowMeat::category { COWMEAT} [static], [private]
```

Kategori dari **CowMeat** (p. 23)

4.10.2.2 price

```
const int CowMeat::price {200000} [static], [private]
```

Harga dari **CowMeat** (p. 23)

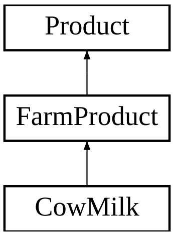
The documentation for this class was generated from the following file:

- **CowMeat.h**

4.11 CowMilk Class Reference

```
#include <CowMilk.h>
```

Inheritance diagram for CowMilk:



Public Member Functions

- int **getPrice** () const
- **Category** **getCategory** () const

Static Private Attributes

- static const int **price** {15000}
- static constexpr **Category** **category** { **COWMEAT**}

4.11.1 Member Function Documentation

4.11.1.1 getCategory()

```
Category CowMilk::getCategory ( ) const  [virtual]
```

Mengembalikan category dari produk

Implements **Product** (p. 66).

4.11.1.2 getPrice()

```
int CowMilk::getPrice ( ) const  [virtual]
```

getPrice mengembalikan harga yang didefinisikan

Implements **Product** (p. 66).

4.11.2 Member Data Documentation

4.11.2.1 category

```
constexpr  Category CowMilk::category { COWMEAT}  [static], [private]
```

Kategori dari **CowMilk** (p. 24)

4.11.2.2 price

```
const int CowMilk::price {15000}  [static], [private]
```

Harga dari **CowMilk** (p. 24)

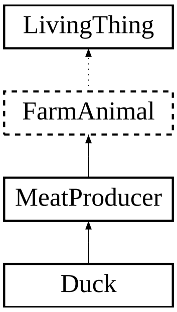
The documentation for this class was generated from the following file:

- **CowMilk.h**

4.12 Duck Class Reference

```
#include <Duck.h>
```

Inheritance diagram for Duck:



Public Member Functions

- **Duck** (**Point** **position**, **Cell** ∗∗∗& **worldMap**, int **nRowCell**, int **nCollumnCell**)
- **FarmProduct** ∗ **ProduceProduct** (**Action**) const
- std::string **makeNoise** () const

Static Private Attributes

- static constexpr int **maxTimeToGetHungryDuck** {15}

4.12.1 Constructor & Destructor Documentation

4.12.1.1 Duck()

```
Duck::Duck (
              Point position,
              Cell ***& worldMap,
             int nRowCell,
             int nCollumnCell )
```

Constructor

4.12.2 Member Function Documentation

4.12.2.1 makeNoise()

```
std::string Duck::makeNoise ( ) const  [virtual]
```

Mengembalikan suara dari **Duck** (p. 26)

Implements **FarmAnimal** (p. 34).

4.12.2.2 ProduceProduct()

```
 FarmProduct* Duck::ProduceProduct (
              Action  ) const
```

Mengembalikan FarmProduk yang akan dihasilkan **Duck** (p. 26) bila **Duck** (p. 26) di kill

4.12.3 Member Data Documentation

4.12.3.1 maxTimeToGetHungryDuck

```
constexpr int Duck::maxTimeToGetHungryDuck {15}  [static], [private]
```

Nilai dari maxTimeToGetHungry

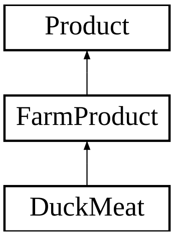
The documentation for this class was generated from the following file:

- **Duck.h**

4.13 DuckMeat Class Reference

```
#include <DuckMeat.h>
```

Inheritance diagram for DuckMeat:



Public Member Functions

- int **getPrice** () const
- **Category** **getCategory** () const

Static Private Attributes

- static const int **price** {25000}
- static constexpr **Category** **category** { **DUCKMEAT**}

4.13.1 Member Function Documentation

4.13.1.1 getCategory()

```
Category DuckMeat::getCategory ( ) const  [virtual]
```

Mengembalikan category dari produk

Implements **Product** (p. 66).

4.13.1.2 getPrice()

```
int DuckMeat::getPrice ( ) const  [virtual]
```

getPrice mengembalikan harga yang didefinisikan

Implements **Product** (p. 66).

4.13.2 Member Data Documentation

4.13.2.1 category

```
constexpr  Category DuckMeat::category { DUCKMEAT}  [static], [private]
```

Kategori dari **DuckMeat** (p. 27)

4.13.2.2 price

```
const int DuckMeat::price {25000}  [static], [private]
```

Harga dari **DuckMeat** (p. 27)

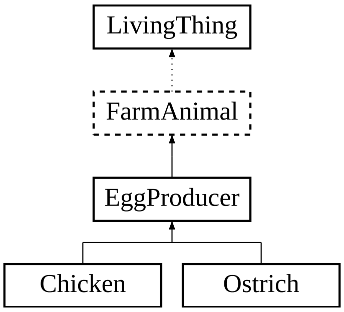
The documentation for this class was generated from the following file:

- **DuckMeat.h**

4.14 EggProducer Class Reference

```
#include <EggProducer.h>
```

Inheritance diagram for EggProducer:



Public Member Functions

- **EggProducer** (int _maxTimeToGetHungry, **Point position**, **Cell** ∗∗∗& **worldMap**, int **nRowCell**, int **nCollumnCell**)
- virtual ∼**EggProducer** ()=0

Private Member Functions

- void **eat** ()
- virtual bool **canMoveTo** (**Cell** toWhere) const

Private Attributes

- bool **canProduce** {false}

4.14.1 Constructor & Destructor Documentation

4.14.1.1 EggProducer()

```
EggProducer::EggProducer (
             int _maxTimeToGetHungry,
             Point position,
             Cell ***& worldMap,
             int nRowCell,
             int nCollumnCell )
```

Constructor maxTimeToGetHungry dengan nilai H

4.14.1.2 ∼EggProducer()

```
virtual EggProducer::∼EggProducer ( )  [pure virtual]
```

Penerusan overloading (virtual) destruktor

4.14.2 Member Function Documentation

4.14.2.1 canMoveTo()

```
virtual bool EggProducer::canMoveTo (
             Cell toWhere ) const  [private], [virtual]
```

Mengecek apakah bisa pindah (tidak out of bound, bertipe **Coop** (p. 20), tidak ada hewan lain)

Reimplemented in **Chicken** (p. 15).

4.14.2.2 eat()

```
void EggProducer::eat ( )  [private], [virtual]
```

Mengubah nilai canProduce

Reimplemented from **FarmAnimal** (p. 34).

4.14.3 Member Data Documentation

4.14.3.1 canProduce

```
bool EggProducer::canProduce {false} [private]
```

Menentukan apakah **FarmAnimal** (p. 32) dapat menghasilkan produk apabila diinteract

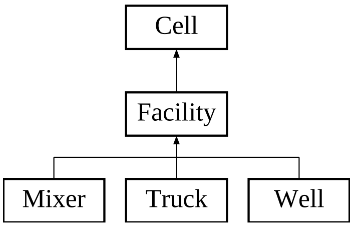
The documentation for this class was generated from the following file:

- **EggProducer.h**

4.15 Facility Class Reference

```
#include <Facility.h>
```

Inheritance diagram for Facility:



Public Member Functions

- virtual ∼**Facility** ()=0
- bool **isFacility** () const
- bool **isGrassExist** () const

Static Private Attributes

- static constexpr bool **facility** {true}

4.15.1 Constructor & Destructor Documentation

4.15.1.1 ∼Facility()

```
virtual Facility::~Facility ( ) [pure virtual]
```

Destructor **Land** (p. 41)

4.15.2 Member Function Documentation

4.15.2.1 isFacility()

```
bool Facility::isFacility ( ) const  [virtual]
```

Return true bila **Land** (p. 41) adalah sebuah facility

Implements **Cell** (p. 13).

4.15.2.2 isGrassExist()

```
bool Facility::isGrassExist ( ) const  [virtual]
```

Mengembalikan false

Implements **Cell** (p. 14).

4.15.3 Member Data Documentation

4.15.3.1 facility

```
constexpr bool Facility::facility {true}  [static], [private]
```

Menandakan bahwa facility

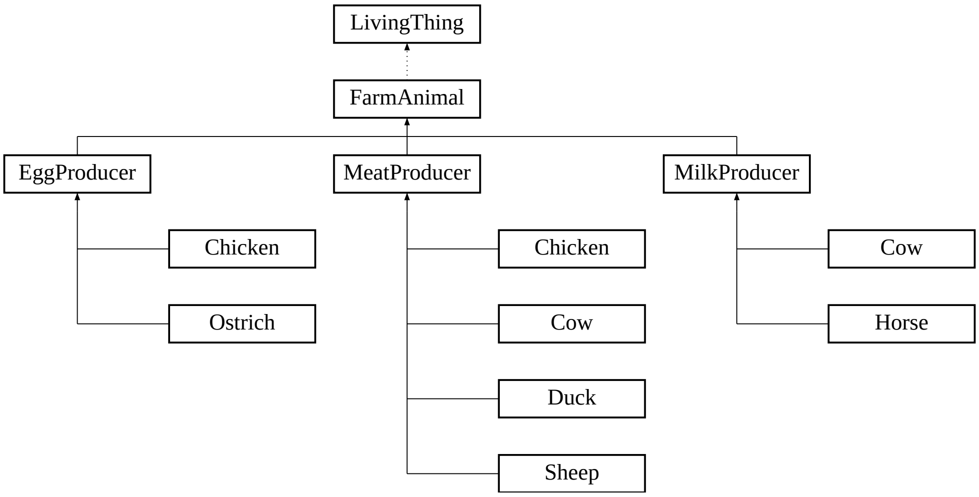
The documentation for this class was generated from the following file:

- **Facility.h**

4.16 FarmAnimal Class Reference

```
#include <FarmAnimal.h>
```

Inheritance diagram for FarmAnimal:



Public Member Functions

- **FarmAnimal** (int _maxTimeToGetHungry, **Point position**, **Cell** ∗∗∗& **worldMap**, int **nRowCell**, int **n**←↩ **CollumnCell**)
- virtual ∼**FarmAnimal** ()=0
- void **tick** ()
- virtual **FarmProduct** ∗ **produceProduct** (**Action**) const =0
- virtual std::string **makeNoise** () const =0

Private Member Functions

- bool **isHungry** () const
- void **decTimeToGetHungry** ()
- void **decTimetoDeath** ()
- bool **isDead** () const
- virtual void **eat** ()
- virtual void **moveRandomly** ()

Private Attributes

- int **timeToGetHungry**
- int **timeToDeath**
- const int **maxTimeToGetHungry**

Static Private Attributes

- static constexpr int **maxTimeToDeath** {5}

4.16.1 Constructor & Destructor Documentation

4.16.1.1 FarmAnimal()

```
FarmAnimal::FarmAnimal (
             int _maxTimeToGetHungry,
              Point position,
              Cell ***& worldMap,
             int nRowCell,
             int nCollumnCell )
```

Constructor maxTimeToGetHungry dengan nilai H

4.16.1.2 ∼FarmAnimal()

```
virtual FarmAnimal::∼FarmAnimal ( )  [pure virtual]
```

Destructor **FarmAnimal** (p. 32)

4.16.2 Member Function Documentation

4.16.2.1 decTimetoDeath()

```
void FarmAnimal::decTimetoDeath ( )  [private]
```

mengurangi timeToDeath

4.16.2.2 decTimeToGetHungry()

```
void FarmAnimal::decTimeToGetHungry ( )  [private]
```

mengurangi timeToGetHungry

4.16.2.3 eat()

```
virtual void FarmAnimal::eat ( )  [private], [virtual]
```

Jika **FarmAnimal** (p. 32) sedang berdiri pada land dengan rumput, maka timeToDeath di set nilai semula dan timeToGdengan nilai sesuai dengan derived classnya, lalu grass di land dihapus

Reimplemented in **EggProducer** (p. 30), and **MilkProducer** (p. 56).

4.16.2.4 isDead()

```
bool FarmAnimal::isDead ( ) const  [private]
```

Mengembalikan true jika timeToDeath == 0, lalu di destruct di main atau di class world

4.16.2.5 isHungry()

```
bool FarmAnimal::isHungry ( ) const  [private]
```

return true apabila timeToGetHungry <= 0

4.16.2.6 makeNoise()

```
virtual std::string FarmAnimal::makeNoise ( ) const  [pure virtual]
```

Mengembalikan suara dari **FarmAnimal** (p. 32)

Implemented in **Chicken** (p. 16), **Cow** (p. 22), **Duck** (p. 26), **Horse** (p. 38), **Ostrich** (p. 59), and **Sheep** (p. 67).

4.16.2.7 moveRandomly()

```
virtual void FarmAnimal::moveRandomly ( )  [private], [virtual]
```

Menggerakan **FarmAnimal** (p. 32) secara random ke posisi yang mungkin ditempati

4.16.2.8 produceProduct()

```
virtual  FarmProduct* FarmAnimal::produceProduct (
              Action  ) const  [pure virtual]
```

Mengembalikan produk yang dihasilkan **FarmAnimal** (p. 32) apabila diinteract/dikill

4.16.2.9 tick()

```
void FarmAnimal::tick ( )
```

Melakukan aksi yang dilakukan **FarmAnimal** (p. 32) setiap satuan waktu

4.16.3 Member Data Documentation

4.16.3.1 maxTimeToDeath

```
constexpr int FarmAnimal::maxTimeToDeath {5}  [static], [private]
```

Nilai max dari timeToDeath

4.16.3.2 maxTimeToGetHungry

```
const int FarmAnimal::maxTimeToGetHungry  [private]
```

Nilai max dari timeToGetHungry

4.16.3.3 timeToDeath

```
int FarmAnimal::timeToDeath  [private]
```

Waktu **FarmAnimal** (p. 32) yang lapar sampai mati Jika tidak lapar, timeToDeath maksimum

4.16.3.4 timeToGetHungry

```
int FarmAnimal::timeToGetHungry  [private]
```

Waktu **FarmAnimal** (p. 32) sampai menjadi lapar

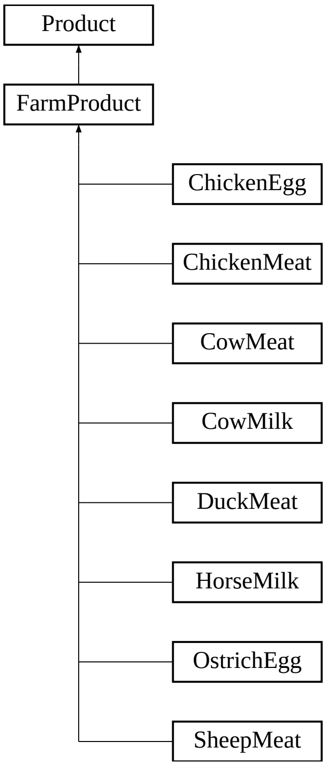
The documentation for this class was generated from the following file:

- **FarmAnimal.h**

4.17 FarmProduct Class Reference

```
#include <FarmProduct.h>
```

Inheritance diagram for FarmProduct:



Additional Inherited Members

4.17.1 Detailed Description

Product (p. 65) yang didapat dari hasil interact / kill

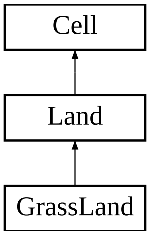
The documentation for this class was generated from the following file:

- **FarmProduct.h**

4.18 GrassLand Class Reference

```
#include <GrassLand.h>
```

Inheritance diagram for GrassLand:



Public Member Functions

- **Category getCategory** () const

Static Private Attributes

- static constexpr **Category category** { **GRASSLAND**}

4.18.1 Member Function Documentation

4.18.1.1 getCategory()

```
Category GrassLand::getCategory ( ) const [virtual]
```

Return kategori dari objek ini

Implements **Cell** (p. 13).

4.18.2 Member Data Documentation

4.18.2.1 category

```
constexpr Category GrassLand::category { GRASSLAND} [static], [private]
```

Menandakan bahwa **Land** (p. 41) ini berkategori **GrassLand** (p. 37)

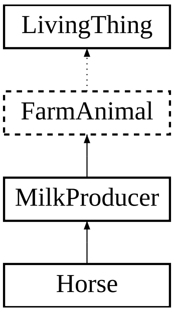
The documentation for this class was generated from the following file:

- **GrassLand.h**

4.19 Horse Class Reference

```
#include <Horse.h>
```

Inheritance diagram for Horse:



Public Member Functions

- **Horse** (**Point position**, **Cell** ∗∗∗& **worldMap**, int **nRowCell**, int **nCollumnCell**)
- **FarmProduct** ∗ **ProduceProduct** (**Action**) const
- std::string **makeNoise** () const

Static Private Attributes

- static constexpr int **maxTimeToGetHungryHorse** {18}

4.19.1 Constructor & Destructor Documentation

4.19.1.1 Horse()

```
Horse::Horse (
              Point position,
              Cell ***& worldMap,
             int nRowCell,
             int nCollumnCell )
```

Constructor

4.19.2 Member Function Documentation

4.19.2.1 makeNoise()

```
std::string Horse::makeNoise ( ) const  [virtual]
```

Mengembalikan suara dari **Horse** (p. 38)

Implements **FarmAnimal** (p. 34).

4.19.2.2 ProduceProduct()

```
 FarmProduct* Horse::ProduceProduct (
             Action  ) const
```

Mengembalikan FarmProduk yang akan dihasilkan **Horse** (p. 38) bila **Horse** (p. 38) di kill

4.19.3 Member Data Documentation

4.19.3.1 maxTimeToGetHungryHorse

```
constexpr int Horse::maxTimeToGetHungryHorse {18}  [static], [private]
```

Nilai dari maxTimeToGetHungry

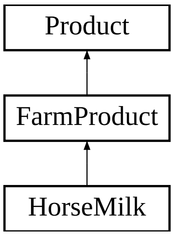
The documentation for this class was generated from the following file:

- **Horse.h**

4.20 HorseMilk Class Reference

```
#include <HorseMilk.h>
```

Inheritance diagram for HorseMilk:



Public Member Functions

- int **getPrice** () const
- **Category** **getCategory** () const

Static Private Attributes

- static const int **price** {35000}
- static constexpr **Category** **category** { **HORSEMILK**}

4.20.1 Member Function Documentation

4.20.1.1 getCategory()

```
Category HorseMilk::getCategory ( ) const [virtual]
```

Mengembalikan category dari produk

Implements **Product** (p. 66).

4.20.1.2 getPrice()

```
int HorseMilk::getPrice ( ) const [virtual]
```

getPrice mengembalikan harga yang didefinisikan

Implements **Product** (p. 66).

4.20.2 Member Data Documentation

4.20.2.1 category

```
constexpr Category HorseMilk::category { HORSEMILK} [static], [private]
```

Kategori dari **HorseMilk** (p. 39)

4.20.2.2 price

```
const int HorseMilk::price {35000}  [static], [private]
```

Harga dari **HorseMilk** (p. 39)

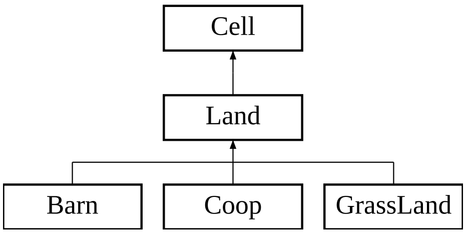
The documentation for this class was generated from the following file:

- **HorseMilk.h**

4.21 Land Class Reference

```
#include <Land.h>
```

Inheritance diagram for Land:



Public Member Functions

- virtual ∼**Land** ()=0
- bool **isFacility** () const
- void **growGrass** ()
- void **removeGrass** ()
- bool **isGrassExist** () const

Private Attributes

- bool **existGrass**

Static Private Attributes

- static constexpr bool **facility** {false}

4.21.1 Constructor & Destructor Documentation

4.21.1.1 ∼Land()

```
virtual Land::∼Land ( )  [pure virtual]
```

Destructor **Land** (p. 41)

4.21.2 Member Function Documentation

4.21.2.1 growGrass()

```
void Land::growGrass ( )  [virtual]
```

Membuat existGrass menjadi true

Reimplemented from **Cell** (p. 13).

4.21.2.2 isFacility()

```
bool Land::isFacility ( ) const  [virtual]
```

Return true bila **Land** (p. 41) adalah sebuah facility

Implements **Cell** (p. 13).

4.21.2.3 isGrassExist()

```
bool Land::isGrassExist ( ) const  [virtual]
```

Mengembalikan keberadaan grass

Implements **Cell** (p. 14).

4.21.2.4 removeGrass()

```
void Land::removeGrass ( )  [virtual]
```

Reimplemented from **Cell** (p. 14).

4.21.3 Member Data Documentation

4.21.3.1 existGrass

```
bool Land::existGrass  [private]
```

Flag yang menandakan apakah terdapat rumput diatas suatu cell atau tidak

4.21.3.2 facility

```
constexpr bool Land::facility {false}  [static], [private]
```

Menandakan bahwa land bukan facility

The documentation for this class was generated from the following file:

- **Land.h**

4.22 LinkedList< T > Class Template Reference

```
#include <LinkedList.h>
```

Public Member Functions

- **LinkedList** ()
- **LinkedList** (std::initializer_list< T > args)
- **LinkedList** (const **LinkedList**< T > &l)
- ∼**LinkedList** ()
- **LinkedList**< T > & **operator=** (const **LinkedList**< T > &l)
- int **find** (T elm)
- bool **isEmpty** () const
- void **add** (T elm)
- void **remove** (T elm)
- void **removeIdx** (int idx)
- T **get** (int idx)
- T & **operator[ ]** (int idx)
- **LinkedList** ()
- **LinkedList** (const **LinkedList**< T > &l)
- ∼**LinkedList** ()
- **LinkedList**< T > & **operator=** (const **LinkedList**< T > &l)
- int **find** (T elm)
- bool **isEmpty** () const
- void **add** (T elm)
- void **remove** (T elm)
- void **removeIdx** (int idx)
- T **get** (int idx)
- T & **operator[ ]** (int idx)

Private Attributes

- **LinkedListNode**< T > * **list**

Friends

- class **LinkedListNode**< **T** >
- ostream & **operator**<< (ostream &os, **LinkedList**< T > l)

4.22.1 Detailed Description

template<class T>
class LinkedList< T >

Tipe data **LinkedList** (p. 43), diimplementasi secara rekursif dengan **LinkedListNode** (p. 49)

4.22.2 Constructor & Destructor Documentation

4.22.2.1 LinkedList() [1/5]

```
template<class T >
LinkedList< T >:: LinkedList ( )
```

Konstruktor default **LinkedList** (p. 43), membuat empty list

4.22.2.2 LinkedList() [2/5]

```
template<class T>
LinkedList< T >:: LinkedList (
            std::initializer_list< T > args )
```

Konstruktor dengan initializer list

4.22.2.3 LinkedList() [3/5]

```
template<class T>
LinkedList< T >:: LinkedList (
            const LinkedList< T > & l )
```

Copy constructor **LinkedList** (p. 43)

4.22.2.4 ∼LinkedList() [1/2]

```
template<class T >
LinkedList< T >::∼ LinkedList ( )
```

Destructor **LinkedList** (p. 43)

4.22.2.5 LinkedList() [4/5]

```
template<class T>
LinkedList< T >::  LinkedList ( )
```

4.22.2.6 LinkedList() [5/5]

```
template<class T>
LinkedList< T >::  LinkedList (
             const  LinkedList< T > & l )
```

4.22.2.7 ∼LinkedList() [2/2]

```
template<class T>
LinkedList< T >::∼ LinkedList ( )
```

4.22.3 Member Function Documentation

4.22.3.1 add() [1/2]

```
template<class T>
void  LinkedList< T >::add (
             T elm )
```

Menambah elm sebagai elemen terakhir

4.22.3.2 add() [2/2]

```
template<class T>
void  LinkedList< T >::add (
             T elm )
```

4.22.3.3 find() [1/2]

```
template<class T>
int LinkedList< T >::find (
             T elm )
```

Mencari indeks pertama dari elm dari **LinkedList** (p. 43). Jika tidak ada, bernilai -1.

4.22.3.4 find() [2/2]

```
template<class T>
int LinkedList< T >::find (
             T elm )
```

4.22.3.5 get() [1/2]

```
template<class T>
T LinkedList< T >::get (
             int idx )
```

4.22.3.6 get() [2/2]

```
template<class T >
T LinkedList< T >::get (
             int idx )
```

Mengembalikan elemen berindeks idx. Jika diluar range, melempar "Index is out of bounds".

4.22.3.7 isEmpty() [1/2]

```
template<class T >
bool LinkedList< T >::isEmpty ( ) const
```

Mengembalikan apakah list empty atau tidak

4.22.3.8 isEmpty() [2/2]

```
template<class T>
bool LinkedList< T >::isEmpty ( ) const
```

4.22.3.9 operator=() [1/2]

```
template<class T>
LinkedList< T > & LinkedList< T >::operator= (
            const LinkedList< T > & l )
```

Operator= **LinkedList** (p. 43)

4.22.3.10 operator=() [2/2]

```
template<class T>
LinkedList<T>& LinkedList< T >::operator= (
            const LinkedList< T > & l )
```

4.22.3.11 operator[]() [1/2]

```
template<class T>
T& LinkedList< T >::operator[] (
            int idx )
```

4.22.3.12 operator[]() [2/2]

```
template<class T >
T & LinkedList< T >::operator[] (
            int idx )
```

Mengembalikan reference ke elemen berindeks idx. Jika diluar range, melempar "Index is out of bounds".

4.22.3.13 remove() [1/2]

```
template<class T>
void LinkedList< T >::remove (
            T elm )
```

Menghapus keberadaan pertama elm Membuat list temp berisi tail untuk dipindahkan ke list sekarang

4.22.3.14 remove() [2/2]

```
template<class T>
void LinkedList< T >::remove (
            T elm )
```

4.22.3.15 removeIdx() [1/2]

```
template<class T>
void  LinkedList< T >::removeIdx (
             int idx )
```

4.22.3.16 removeIdx() [2/2]

```
template<class T >
void  LinkedList< T >::removeIdx (
             int idx )
```

Menghapus elemen berindeks idx. Jika diluar range, melempar "Index is out of bounds". Membuat list temp berisi tail untuk dipindahkan ke list sekarang

4.22.4 Friends And Related Function Documentation

4.22.4.1 LinkedListNode< T >

```
template<class T>
friend class  LinkedListNode< T >  [friend]
```

4.22.4.2 operator<<

```
template<class T>
ostream& operator<< (
             ostream & os,
              LinkedList< T > l )  [friend]
```

4.22.5 Member Data Documentation

4.22.5.1 list

```
template<class T>
LinkedListNode< T > *  LinkedList< T >::list  [private]
```

Pointer ke **LinkedListNode** (p. 49), kalau empty bernilai nullptr

The documentation for this class was generated from the following files:

- **LinkedList.h**
- **LinkedListDriver.cpp**

4.23 LinkedListNode< T > Class Template Reference

```
#include <LinkedList.h>
```

Public Member Functions

- **LinkedListNode** (T **head**, **LinkedList**< T > **tail**)
- **LinkedListNode** (T **head**, **LinkedList**< T > **tail**)

Public Attributes

- friend **LinkedList**< **T** >

Private Attributes

- T **head**
- **LinkedList**< T > **tail**

Friends

- class **LinkedList**< **T** >

4.23.1 Detailed Description

template< class T>
class LinkedListNode< T >

Forward declaration dari kelas **LinkedListNode** (p. 49)

Anggota kelas implementasi **LinkedList** (p. 43) secara rekursifs

4.23.2 Constructor & Destructor Documentation

4.23.2.1 LinkedListNode() [1/2]

```
template<class T>
LinkedListNode< T >:: LinkedListNode (
              T head,
               LinkedList< T > tail )
```

Konstruktor **LinkedListNode** (p. 49) dengan initializer list

4.23.2.2 LinkedListNode() [2/2]

```
template<class T>
LinkedListNode< T >:: LinkedListNode (
             T head,
              LinkedList< T > tail )
```

4.23.3 Friends And Related Function Documentation

4.23.3.1 LinkedList< T >

```
template<class T>
friend class  LinkedList< T >  [friend]
```

4.23.4 Member Data Documentation

4.23.4.1 head

```
template<class T>
T  LinkedListNode< T >::head  [private]
```

Tipe data pertama pada **LinkedListNode** (p. 49)

4.23.4.2 LinkedList< T >

```
template<class T>
friend  LinkedListNode< T >::  LinkedList< T >
```

Membuat **LinkedList** (p. 43) dapat mengakses head dan tail

4.23.4.3 tail

```
template<class T>
LinkedList< T >  LinkedListNode< T >::tail  [private]
```

Sisa dari **LinkedListNode** (p. 49) berupa **LinkedList** (p. 43)

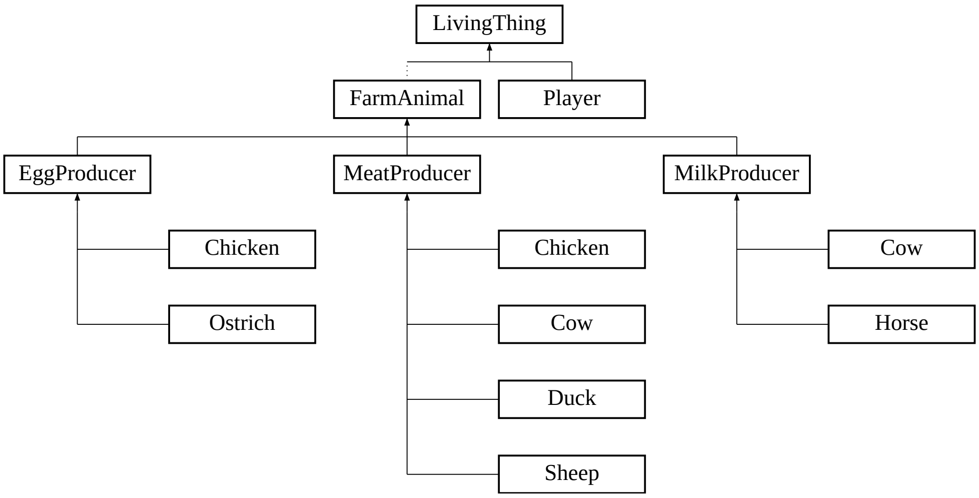
The documentation for this class was generated from the following files:

- **LinkedList.h**
- **LinkedListDriver.cpp**

4.24 LivingThing Class Reference

```
#include <LivingThing.h>
```

Inheritance diagram for LivingThing:



Public Member Functions

- **LivingThing** (**Point** **position**, **Cell** ∗∗∗& **worldMap**, int **nRowCell**, int **nCollumnCell**)
- virtual ∼**LivingThing** ()=0
- **Point** **getPosition** () const
- bool **move** (Direction toWhere)
- virtual char **render** ()=0

Protected Attributes

- **Cell** ∗∗∗& **worldMap**
- int **nRowCell**
- int **nCollumnCell**

Private Member Functions

- virtual bool **canMoveTo** (**Cell** toWhere)=0

Private Attributes

- **Point** **position**

4.24.1 Constructor & Destructor Documentation

4.24.1.1 LivingThing()

```
LivingThing::LivingThing (
              Point position,
              Cell ***& worldMap,
             int nRowCell,
             int nCollumnCell )
```

Constructor **LivingThing** (p. 51)

4.24.1.2 ∼LivingThing()

```
virtual LivingThing::∼LivingThing ( )  [pure virtual]
```

Destructor dari **LivingThing** (p. 51)

4.24.2 Member Function Documentation

4.24.2.1 canMoveTo()

```
virtual bool LivingThing::canMoveTo (
              Cell toWhere )  [private], [pure virtual]
```

Apakah bisa masuk suatu area (cek out of bound, jenis **Cell** (p. 12), kekosongan **Cell** (p. 12))

Implemented in **Player** (p. 62).

4.24.2.2 getPosition()

```
 Point LivingThing::getPosition ( ) const
```

Mengembalikan position

4.24.2.3 move()

```
bool LivingThing::move (
             Direction toWhere )
```

Berpindah ke suatu lokasi. Apabila tidak bisa (!canMoveTo), throw "Cannot move to the direction".

4.24.2.4 render()

```
virtual char LivingThing::render ( )  [pure virtual]
```

Mengembalikan char untuk direnderkan ke layar

Implemented in **Player** (p. 63).

4.24.3 Member Data Documentation

4.24.3.1 nCollumnCell

```
int LivingThing::nCollumnCell  [protected]
```

Nilai efektif kolom untuk Matriks **Cell** (p. 12)

4.24.3.2 nRowCell

```
int LivingThing::nRowCell  [protected]
```

Nilai efektif baris untuk Matriks **Cell** (p. 12)

4.24.3.3 position

```
Point LivingThing::position  [private]
```

Posisi dari **LivingThing** (p. 51)

4.24.3.4 worldMap

```
Cell***& LivingThing::worldMap  [protected]
```

Representasi dunia tempat **LivingThing** (p. 51) tinggal

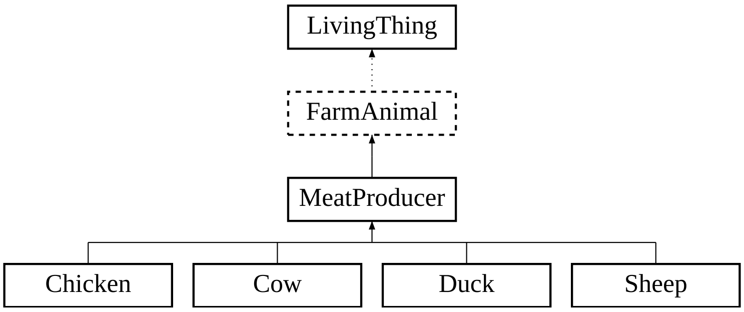
The documentation for this class was generated from the following file:

- **LivingThing.h**

4.25 MeatProducer Class Reference

```
#include <MeatProducer.h>
```

Inheritance diagram for MeatProducer:



Public Member Functions

- **MeatProducer** (int _maxTimeToGetHungry, **Point position**, **Cell** ∗∗∗& **worldMap**, int **nRowCell**, int **nCollumnCell**)
- virtual ∼**MeatProducer** ()=0

Private Member Functions

- virtual bool **canMoveTo** (**Cell** toWhere) const

4.25.1 Constructor & Destructor Documentation

4.25.1.1 MeatProducer()

```
MeatProducer::MeatProducer (
             int _maxTimeToGetHungry,
              Point position,
              Cell ***& worldMap,
             int nRowCell,
             int nCollumnCell )
```

Constructor maxTimeToGetHungry dengan nilai H

4.25.1.2 ∼MeatProducer()

```
virtual MeatProducer::∼MeatProducer ( )  [pure virtual]
```

Penerusan overloading (virtual) destruktor

4.25.2 Member Function Documentation

4.25.2.1 canMoveTo()

```
virtual bool MeatProducer::canMoveTo (
              Cell toWhere ) const  [private], [virtual]
```

Mengecek apakah bisa pindah (tidak out of bound, bertipe **GrassLand** (p. 37), tidak ada hewan lain)

Reimplemented in **Chicken** (p. 15), and **Cow** (p. 21).

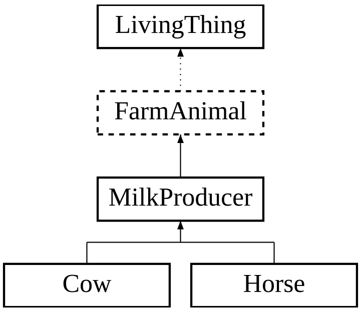
The documentation for this class was generated from the following file:

- **MeatProducer.h**

4.26 MilkProducer Class Reference

```
#include <MilkProducer.h>
```

Inheritance diagram for MilkProducer:



Public Member Functions

- **MilkProducer** (int _maxTimeToGetHungry, **Point position**, **Cell** ∗∗∗& **worldMap**, int **nRowCell**, int **n**↩ **CollumnCell**)
- virtual ∼**MilkProducer** ()=0

Private Member Functions

- void **eat** ()
- virtual bool **canMoveTo** (**Cell** toWhere) const

Private Attributes

- bool **canProduce** {false}

4.26.1 Constructor & Destructor Documentation

4.26.1.1 MilkProducer()

```
MilkProducer::MilkProducer (
             int _maxTimeToGetHungry,
              Point position,
              Cell ***& worldMap,
             int nRowCell,
             int nCollumnCell )
```

Constructor maxTimeToGetHungry dengan nilai H

4.26.1.2 ∼MilkProducer()

```
virtual MilkProducer::∼MilkProducer ( )  [pure virtual]
```

Penerusan overloading (virtual) destruktor

4.26.2 Member Function Documentation

4.26.2.1 canMoveTo()

```
virtual bool MilkProducer::canMoveTo (
              Cell toWhere ) const  [private], [virtual]
```

Mengecek apakah bisa pindah (tidak out of bound, bertipe **Barn** (p. 7), tidak ada hewan lain)

Reimplemented in **Cow** (p. 21).

4.26.2.2 eat()

```
void MilkProducer::eat ( )  [private], [virtual]
```

Mengubah nilai canProduce

Reimplemented from **FarmAnimal** (p. 34).

4.26.3 Member Data Documentation

4.26.3.1 canProduce

```
bool MilkProducer::canProduce {false}  [private]
```

Menentukan apakah **FarmAnimal** (p. 32) dapat menghasilkan produk apabila diinteract

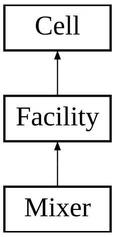
The documentation for this class was generated from the following file:

- **MilkProducer.h**

4.27 Mixer Class Reference

```
#include <Mixer.h>
```

Inheritance diagram for Mixer:



Public Member Functions

- **Category getCategory** () const

Static Private Attributes

- static constexpr **Category category** { **MIXER**}

4.27.1 Member Function Documentation

4.27.1.1 getCategory()

```
Category Mixer::getCategory ( ) const  [virtual]
```

Return kategori dari objek ini

Implements **Cell** (p. 13).

4.27.2 Member Data Documentation

4.27.2.1 category

```
constexpr  Category Mixer::category { MIXER}  [static], [private]
```

Menandakan bahwa land bertipe **Mixer** (p. 57)

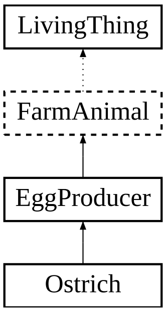
The documentation for this class was generated from the following file:

- **Mixer.h**

4.28 Ostrich Class Reference

```
#include <Ostrich.h>
```

Inheritance diagram for Ostrich:



Public Member Functions

- **Ostrich** (**Point position**, **Cell** ∗∗∗& **worldMap**, int **nRowCell**, int **nCollumnCell**)
- **FarmProduct** ∗ **ProduceProduct** (**Action**) const
- std::string **makeNoise** () const

Static Private Attributes

- static constexpr int **maxTimeToGetHungryOstrich** {15}

4.28.1 Constructor & Destructor Documentation

4.28.1.1 Ostrich()

```
Ostrich::Ostrich (
               Point position,
               Cell ***& worldMap,
              int nRowCell,
              int nCollumnCell )
```

Constructor

4.28.2 Member Function Documentation

4.28.2.1 makeNoise()

```
std::string Ostrich::makeNoise ( ) const  [virtual]
```

Mengembalikan suara dari **Chicken** (p. 15)

Implements **FarmAnimal** (p. 34).

4.28.2.2 ProduceProduct()

```
 FarmProduct* Ostrich::ProduceProduct (
              Action  ) const
```

Mengembalikan FarmProduk yang akan dihasilkan **Ostrich** (p. 58) bila **Ostrich** (p. 58) di kill

4.28.3 Member Data Documentation

4.28.3.1 maxTimeToGetHungryOstrich

```
constexpr int Ostrich::maxTimeToGetHungryOstrich {15}  [static], [private]
```

Nilai dari maxTimeToGetHungry

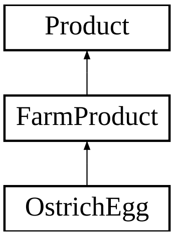
The documentation for this class was generated from the following file:

- **Ostrich.h**

4.29 OstrichEgg Class Reference

```
#include <OstrichEgg.h>
```

Inheritance diagram for OstrichEgg:



Public Member Functions

- int **getPrice** () const
- **Category** **getCategory** () const

Static Private Attributes

- static const int **price** {40000}
- static constexpr **Category** **category** { **OSTRICHEGG**}

4.29.1 Member Function Documentation

4.29.1.1 getCategory()

```
Category OstrichEgg::getCategory ( ) const [virtual]
```

Mengembalikan category dari produk

Implements **Product** (p. 66).

4.29.1.2 getPrice()

```
int OstrichEgg::getPrice ( ) const [virtual]
```

getPrice mengembalikan harga yang didefinisikan

Implements **Product** (p. 66).

4.29.2 Member Data Documentation

4.29.2.1 category

```
constexpr  Category OstrichEgg::category { OSTRICHEGG}  [static], [private]
```

Kategori dari **OstrichEgg** (p. 60)

4.29.2.2 price

```
const int OstrichEgg::price {40000}  [static], [private]
```

Harga dari **OstrichEgg** (p. 60)

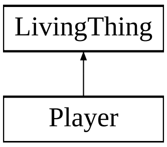
The documentation for this class was generated from the following file:

- **OstrichEgg.h**

4.30 Player Class Reference

```
#include <Player.h>
```

Inheritance diagram for Player:



Public Member Functions

- **Player** (**Point position**, **Cell** $***$& **worldMap**, int **nRowCell**, int **nCollumnCell**)
- $\sim$**Player** ()
- void **talk** (**LinkedList**$<$ **FarmAnimal** $>$ &farmAnimal)
- void **interact** (**LinkedList**$<$ **FarmAnimal** $>$ &farmAnimal)
- void **kill** (**LinkedList**$<$ **FarmAnimal** $>$ &farmAnimal)
- void **grow** ()
- void **mix** (**Product** $*$makeTo)
- char **render** ()

Private Member Functions

- bool **canMoveTo** (**Cell** toWhere)

Private Attributes

- **LinkedList**< **Product** & > **inventory**
- int **money** {500000}
- int **water** {5}

Static Private Attributes

- static **LinkedList**< **SideProduct** ∗ > **recipeBook**

Additional Inherited Members

4.30.1 Constructor & Destructor Documentation

4.30.1.1 Player()

```
Player::Player (
              Point position,
              Cell ***& worldMap,
             int nRowCell,
             int nCollumnCell )
```

Constructor **Player** (p. 61) di position, recipeBook diinisalisasi dengan semua **SideProduct** (p. 69) yang terdefinisi

4.30.1.2 ∼Player()

```
Player::∼Player ( )
```

Destructor **Player** (p. 61)

4.30.2 Member Function Documentation

4.30.2.1 canMoveTo()

```
bool Player::canMoveTo (
              Cell toWhere )  [private], [virtual]
```

Apakah bisa masuk suatu area (cek out of bound, jenis **Cell** (p. 12), kekosongan **Cell** (p. 12))

Implements **LivingThing** (p. 52).

4.30.2.2 grow()

```
void Player::grow ( )
```

Menumbuhkan rumput pada cell yang sedang ditempati oleh **Player** (p. 61)

4.30.2.3 interact()

```
void Player::interact (
              LinkedList< FarmAnimal > & farmAnimal )
```

Player (p. 61) mengambil **FarmProduct** (p. 36) dari semua **FarmAnimal** (p. 32) terdekat tanpa membunuh **FarmAnimal** (p. 32) tersebut. Bekerja untuk **FarmAnimal** (p. 32) jenis MilkProducing dan EggProducing. Contoh **FarmProduct** (p. 36) : **ChickenEgg** (p. 17), **CowMilk** (p. 24).

4.30.2.4 kill()

```
void Player::kill (
              LinkedList< FarmAnimal > & farmAnimal )
```

Player (p. 61) mengambil **FarmProduct** (p. 36) dari semua **FarmAnimal** (p. 32) terdekat dengan cara membunuh **FarmAnimal** (p. 32) tersebut. Bekerja untuk **FarmAnimal** (p. 32) jenis MeatProducing. Contoh **FarmProduct** (p. 36) : **CowMeat** (p. 23), **ChickenMeat** (p. 18).

4.30.2.5 mix()

```
void Player::mix (
              Product * makeTo )
```

Menciptakan **SideProduct** (p. 69) dari **FarmProduct** (p. 36) bila **Player** (p. 61) dekat dengan mixer

4.30.2.6 render()

```
char Player::render ( ) [virtual]
```

Mengembalikan char untuk dirender ke layar

Implements **LivingThing** (p. 52).

4.30.2.7 talk()

```
void Player::talk (
              LinkedList< FarmAnimal > & farmAnimal )
```

Player (p. 61) berbicara dengan semua **FarmAnimal** (p. 32) terdekat.

4.30.3 Member Data Documentation

4.30.3.1 inventory

```
LinkedList< Product&> Player::inventory [private]
```

Product (p. 65) yang dipegang **Player** (p. 61)

4.30.3.2 money

```
int Player::money {500000} [private]
```

Uang yang dimiliki **Player** (p. 61)

4.30.3.3 recipeBook

```
LinkedList< SideProduct*> Player::recipeBook [static], [private]
```

Digunakan untuk melakukan pengecekan saat melakukan method mix Contoh Pengunaan : Bila player ingin membuat **BeefMuttonSate** (p. 10), program tranversal di recipeeBook sampai menemukan sideProdect dengan Category = BEEFMUTTONSATE lalu melihat resep dari objek tersebut. recipeBook diinisalisasi di implementasi

4.30.3.4 water

```
int Player::water {5} [private]
```

Air yang dipegang **Player** (p. 61)

The documentation for this class was generated from the following file:

- **Player.h**

4.31 Point Struct Reference

```
#include <Point.h>
```

Public Attributes

- int **x**
- int **y**

4.31.1 Member Data Documentation

4.31.1.1 x

```
int Point::x
```

4.31.1.2 y

```
int Point::y
```

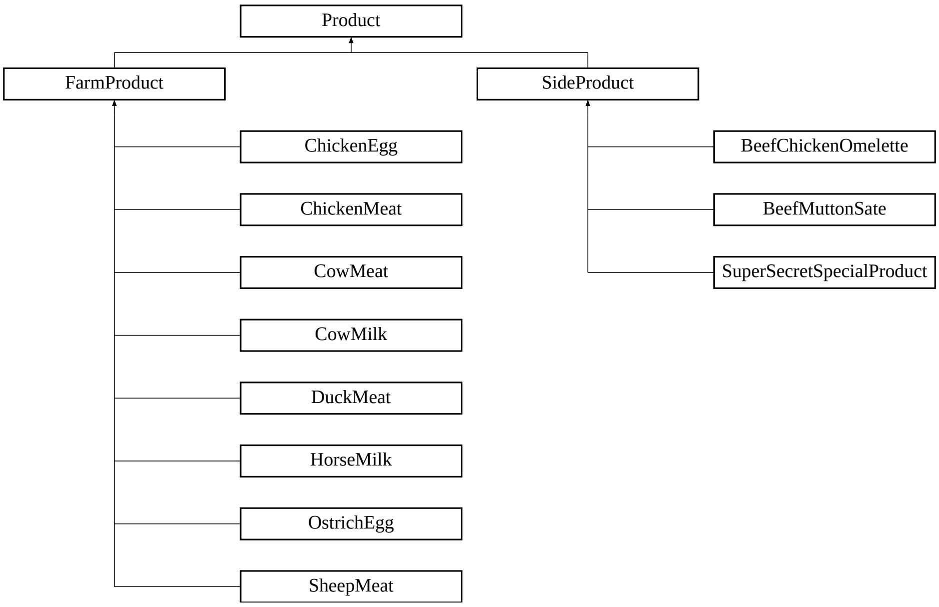
The documentation for this struct was generated from the following file:

- **Point.h**

4.32 Product Class Reference

```
#include <Product.h>
```

Inheritance diagram for Product:



Public Member Functions

- virtual int **getPrice** () const =0
- virtual **Category** **getCategory** () const =0

4.32.1 Member Function Documentation

4.32.1.1 getCategory()

```
virtual  Category Product::getCategory ( ) const  [pure virtual]
```

mengembalikan kategori dari produk ini

Implemented in **BeefChickenOmelette** (p. 9), **BeefMuttonSate** (p. 11), **SuperSecretSpecialProduct** (p. 71), **ChickenEgg** (p. 17), **ChickenMeat** (p. 19), **CowMeat** (p. 23), **CowMilk** (p. 25), **DuckMeat** (p. 28), **HorseMilk** (p. 40), **OstrichEgg** (p. 60), and **SheepMeat** (p. 68).

4.32.1.2 getPrice()

```
virtual int Product::getPrice ( ) const  [pure virtual]
```

getPrice mengembalikan harga yang didefinisikan

Implemented in **BeefChickenOmelette** (p. 9), **BeefMuttonSate** (p. 11), **SuperSecretSpecialProduct** (p. 71), **ChickenEgg** (p. 17), **ChickenMeat** (p. 19), **CowMeat** (p. 23), **CowMilk** (p. 25), **DuckMeat** (p. 28), **HorseMilk** (p. 40), **OstrichEgg** (p. 60), and **SheepMeat** (p. 69).

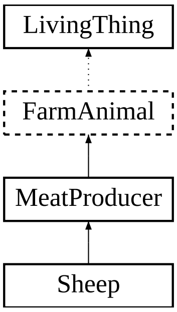
The documentation for this class was generated from the following file:

- **Product.h**

4.33 Sheep Class Reference

```
#include <Sheep.h>
```

Inheritance diagram for Sheep:



Public Member Functions

- **Sheep** (**Point position**, **Cell** ∗∗∗& **worldMap**, int **nRowCell**, int **nCollumnCell**)
- **FarmProduct** ∗ **ProduceProduct** (**Action**) const
- std::string **makeNoise** () const

Static Private Attributes

- static constexpr int **maxTimeToGetHungrySheep** {15}

4.33.1 Constructor & Destructor Documentation

4.33.1.1 Sheep()

```
Sheep::Sheep (
              Point position,
              Cell ∗∗∗& worldMap,
             int nRowCell,
             int nCollumnCell )
```

Constructor

4.33.2 Member Function Documentation

4.33.2.1 makeNoise()

```
std::string Sheep::makeNoise ( ) const  [virtual]
```

Mengembalikan suara dari **Sheep** (p. 66)

Implements **FarmAnimal** (p. 34).

4.33.2.2 ProduceProduct()

```
 FarmProduct∗ Sheep::ProduceProduct (
              Action  ) const
```

Mengembalikan FarmProduk yang akan dihasilkan **Sheep** (p. 66) bila **Sheep** (p. 66) di kill

4.33.3 Member Data Documentation

4.33.3.1 maxTimeToGetHungrySheep

```
constexpr int Sheep::maxTimeToGetHungrySheep {15}  [static], [private]
```

Nilai dari maxTimeToGetHungry

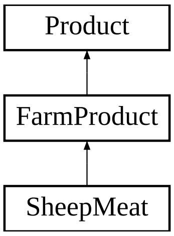
The documentation for this class was generated from the following file:

- **Sheep.h**

4.34 SheepMeat Class Reference

```
#include <SheepMeat.h>
```

Inheritance diagram for SheepMeat:



Public Member Functions

- int **getPrice** () const
- **Category getCategory** () const

Static Private Attributes

- static const int **price** {100000}
- static constexpr **Category category** { **SHEEPMEAT**}

4.34.1 Member Function Documentation

4.34.1.1 getCategory()

```
Category SheepMeat::getCategory ( ) const  [virtual]
```

Mengembalikan category dari produk

Implements **Product** (p. 66).

4.34.1.2 getPrice()

```
int SheepMeat::getPrice ( ) const  [virtual]
```

getPrice mengembalikan harga yang didefinisikan

Implements **Product** (p. 66).

4.34.2 Member Data Documentation

4.34.2.1 category

```
constexpr  Category SheepMeat::category { SHEEPMEAT}  [static], [private]
```

Kategori dari **SheepMeat** (p. 68)

4.34.2.2 price

```
const int SheepMeat::price {100000}  [static], [private]
```

Harga dari **SheepMeat** (p. 68)

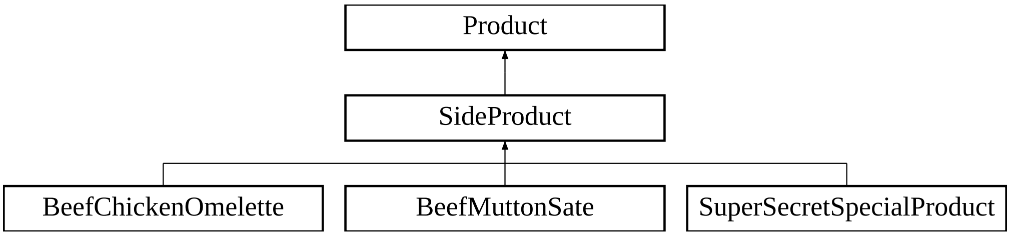
The documentation for this class was generated from the following file:

- **SheepMeat.h**

4.35 SideProduct Class Reference

```
#include <SideProduct.h>
```

Inheritance diagram for SideProduct:



Additional Inherited Members

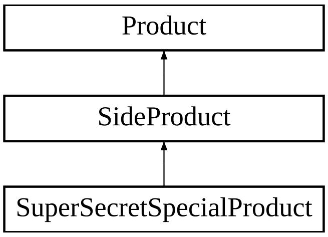
The documentation for this class was generated from the following file:

- **SideProduct.h**

4.36 SuperSecretSpecialProduct Class Reference

```
#include <SuperSecretSpecialProduct.h>
```

Inheritance diagram for SuperSecretSpecialProduct:



Public Member Functions

- **SuperSecretSpecialProduct** ()
- int **getPrice** () const
- **Category getCategory** () const

Static Public Member Functions

- static **LinkedList**< **Product** ∗ > & **getRecipe** ()

Static Private Attributes

- static const int **price** {999999}
- static constexpr **Category category** { **SUPERSECRETSPECIALPRODUCT**}
- static const **LinkedList**< **Product** ∗ > **recipe**

4.36.1 Constructor & Destructor Documentation

4.36.1.1 SuperSecretSpecialProduct()

```
SuperSecretSpecialProduct::SuperSecretSpecialProduct ( )
```

Constructor untuk inisialisasi recipe

4.36.2 Member Function Documentation

4.36.2.1 getCategory()

```
Category SuperSecretSpecialProduct::getCategory ( ) const [virtual]
```

Mengembalikan category dari produk

Implements **Product** (p. 66).

4.36.2.2 getPrice()

```
int SuperSecretSpecialProduct::getPrice ( ) const [virtual]
```

getPrice mengembalikan harga yang didefinisikan

Implements **Product** (p. 66).

4.36.2.3 getRecipe()

```
static LinkedList< Product*>& SuperSecretSpecialProduct::getRecipe ( ) [static]
```

Mengembalikan resep dari produk

4.36.3 Member Data Documentation

4.36.3.1 category

```
constexpr Category SuperSecretSpecialProduct::category { SUPERSECRETSPECIALPRODUCT} [static],
[private]
```

Kategori dari **SuperSecretSpecialProduct** (p. 70)

4.36.3.2 price

```
const int SuperSecretSpecialProduct::price {999999} [static], [private]
```

Harga dari **SuperSecretSpecialProduct** (p. 70)

4.36.3.3 recipe

```
const LinkedList< Product*> SuperSecretSpecialProduct::recipe [static], [private]
```

Resep **SuperSecretSpecialProduct** (p. 70). Terdiri dari **OstrichEgg** (p. 60) dan HorseMeat.

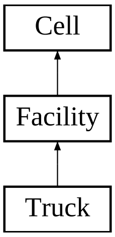
The documentation for this class was generated from the following file:

- **SuperSecretSpecialProduct.h**

4.37 Truck Class Reference

```
#include <Truck.h>
```

Inheritance diagram for Truck:



Public Member Functions

- **Category getCategory** () const

Static Private Attributes

- static constexpr **Category category** { **TRUCK**}

4.37.1 Member Function Documentation

4.37.1.1 getCategory()

```
Category Truck::getCategory ( ) const [virtual]
```

Return kategori dari objek ini

Implements **Cell** (p. 13).

4.37.2 Member Data Documentation

4.37.2.1 category

`constexpr **Category** Truck::category { **TRUCK**} [static], [private]`

Menandakan bahwa land bertipe **Truck** (p. 72)

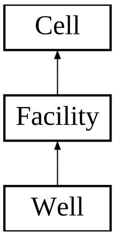
The documentation for this class was generated from the following file:

- **Truck.h**

4.38 Well Class Reference

`#include <Well.h>`

Inheritance diagram for Well:



Public Member Functions

- **Category getCategory** () const

Static Private Attributes

- static constexpr **Category category** { **WELL**}

4.38.1 Member Function Documentation

4.38.1.1 getCategory()

`**Category** Well::getCategory ( ) const [virtual]`

Return kategori dari objek ini

Implements **Cell** (p. 13).

4.38.2 Member Data Documentation

4.38.2.1 category

```
constexpr  Category Well::category { WELL}  [static], [private]
```

Menandakan bahwa land bertipe **Well** (p. 73)

The documentation for this class was generated from the following file:

- **Well.h**

4.39 World Class Reference

```
#include <World.h>
```

Public Member Functions

- **World** ()
- ∼**World** ()
- void **Input** ()
- void **Update** ()
- void **Draw** ()

Private Attributes

- **Player pl**
- **Cell** $***$ **map**
- int **nRowCell**
- int **nCollumnCell**
- **LinkedList**$<$ **FarmAnimal** $*$ $>$ **animalList**

4.39.1 Constructor & Destructor Documentation

4.39.1.1 World()

```
World::World ( )
```

Constructor **World** (p. 74). Memanggil ctor dan menginisialisasi semua atribut world; Pertama, map diinisialisasi sesuai dengan spesifikasi, saat penginisialisasian map, ctor untuk object riil dari cell seperti coop, barn, dan well dipanggil Kedua, ctor **Player** (p. 61) dipanggil dengan argumen **Point** (p. 64) lokasi awal player dan reference ke map yang sudah didefinisikan pada tahap pertama Terakhir, animalList diinisialisasi dengan beberapa **FarmAnimal** (p. 32) secara random

4.39.1.2 ∼World()

```
World::∼World ( )
```

Destructor **World** (p. 74). Dealokasi seluruh **Cell** (p. 12) dan **FarmAnimal** (p. 32), termasuk seluruh pointer yang berhubungan.

4.39.2 Member Function Documentation

4.39.2.1 Draw()

```
void World::Draw ( )
```

Megambarkar representasi state program (**World** (p. 74)) seperti lokasi setiap objek, money, water, dan Inventory **Player** (p. 61), dsb ke layar.

4.39.2.2 Input()

```
void World::Input ( )
```

Membaca input user dari stdin lalu melakukan aksi sesuai degan spesifikasi, misal, input == MOVELEFT, maka akan dipanggil pl.move(LEFT). Bila input == INTERACT, maka akan dipanggil pl.interact(animalList), dsb.

4.39.2.3 Update()

```
void World::Update ( )
```

Pada **World::Update()** (p. 75), setiap fungsi yang dipanggil secara berkala seperti **FarmAnimal::tick()** (p. 35) akan dipanggil.

4.39.3 Member Data Documentation

4.39.3.1 animalList

```
LinkedList< FarmAnimal*> World::animalList [private]
```

LinkedList (p. 43) dari seluruh pointer ke **FarmAnimal** (p. 32) yang berada pada **World** (p. 74) 000

4.39.3.2 map

```
Cell*** World::map [private]
```

Matriks dari pointer ke seluruh **Cell** (p. 12) pada **World** (p. 74)

4.39.3.3 nCollumnCell

```
int World::nCollumnCell [private]
```

Nilai efektif kolom untuk Matriks **Cell** (p. 12)

4.39.3.4 nRowCell

```
int World::nRowCell [private]
```

Nilai efektif baris untuk Matriks **Cell** (p. 12)

4.39.3.5 pl

```
Player World::pl [private]
```

Player (p. 61) yang berada pada **World** (p. 74)

The documentation for this class was generated from the following file:

- **World.h**

Chapter 5

File Documentation

5.1 Barn.h File Reference

```
#include "Land.h"
```

Classes

- class **Barn**

5.2 BeefChickenOmelette.h File Reference

```
#include "../LinkedList.h"
#include "SideProduct.h"
```

Classes

- class **BeefChickenOmelette**

5.3 BeefMuttonSate.h File Reference

```
#include "../LinkedList.h"
#include "SideProduct.h"
```

Classes

- class **BeefMuttonSate**

5.4 Cell.h File Reference

Classes

- class **Cell**

Macros

- #define **Category** unsigned int
- #define **WELL** 17
- #define **MIXER** 18
- #define **TRUCK** 19
- #define **COOP** 20
- #define **GRASSLAND** 21
- #define **BARN** 22

5.4.1 Macro Definition Documentation

5.4.1.1 BARN

```
#define BARN 22
```

5.4.1.2 Category

```
#define Category unsigned int
```

5.4.1.3 COOP

```
#define COOP 20
```

5.4.1.4 GRASSLAND

```
#define GRASSLAND 21
```

5.4.1.5 MIXER

```
#define MIXER 18
```

5.4.1.6 TRUCK

```
#define TRUCK 19
```

5.4.1.7 WELL

```
#define WELL 17
```

5.5 Chicken.h File Reference

```
#include "../Point.h"
#include "../Cell/Cell.h"
#include "EggProducer.h"
#include "MeatProducer.h"
#include <string>
```

Classes

- class **Chicken**

5.6 ChickenEgg.h File Reference

```
#include "FarmProduct.h"
```

Classes

- class **ChickenEgg**

5.7 ChickenMeat.h File Reference

```
#include "FarmProduct.h"
```

Classes

- class **ChickenMeat**

5.8 Coop.h File Reference

```
#include "Land.h"
```

Classes

- class **Coop**

5.9 Cow.h File Reference

```
#include "../Point.h"
#include "../Cell/Cell.h"
#include "MilkProducer.h"
#include "MeatProducer.h"
#include <string>
```

Classes

- class **Cow**

5.10 CowMeat.h File Reference

```
#include "FarmProduct.h"
```

Classes

- class **CowMeat**

5.11 CowMilk.h File Reference

```
#include "FarmProduct.h"
```

Classes

- class **CowMilk**

5.12 Direction.h File Reference

Macros

- #define **DIRECTION** unsigned int
- #define **LEFT** 0
- #define **RIGHT** 1
- #define **UP** 2
- #define **DOWN** 3

5.12.1 Macro Definition Documentation

5.12.1.1 DIRECTION

```
#define DIRECTION unsigned int
```

5.12.1.2 DOWN

```
#define DOWN 3
```

5.12.1.3 LEFT

```
#define LEFT 0
```

5.12.1.4 RIGHT

```
#define RIGHT 1
```

5.12.1.5 UP

```
#define UP 2
```

5.13 Duck.h File Reference

```
#include "../Point.h"
#include "../Cell/Cell.h"
#include "MeatProducer.h"
#include <string>
```

Classes

- class **Duck**

5.14 DuckMeat.h File Reference

```
#include "FarmProduct.h"
```

Classes

- class **DuckMeat**

Macros

- #define **DUCk_MEAT_H**

5.14.1 Macro Definition Documentation

5.14.1.1 DUCk_MEAT_H

```
#define DUCk_MEAT_H
```

5.15 EggProducer.h File Reference

```
#include "../Point.h"
#include "../Cell/Cell.h"
#include "FarmAnimal.h"
```

Classes

- class **EggProducer**

5.16 Facility.h File Reference

```
#include "Cell.h"
```

Classes

- class **Facility**

5.17 FarmAnimal.h File Reference

```
#include "../Point.h"
#include "../Cell/Cell.h"
#include "../LivingThing.h"
#include "../Product/FarmProduct.h"
#include <string>
```

Classes

- class **FarmAnimal**

Macros

- #define **Action** unsigned int
- #define **INTERACT** 4
- #define **KILL** 5

5.17.1 Macro Definition Documentation

5.17.1.1 Action

```
#define Action unsigned int
```

Jenis aksi yang dapat dilakukan ke **FarmAnimal** (p. 32)

5.17.1.2 INTERACT

```
#define INTERACT 4
```

5.17.1.3 KILL

```
#define KILL 5
```

5.18 FarmProduct.h File Reference

```
#include "Product.h"
```

Classes

- class **FarmProduct**

5.19 GrassLand.h File Reference

```
#include "Land.h"
```

Classes

- class **GrassLand**

5.20 Horse.h File Reference

```
#include "../Point.h"
#include "../Cell/Cell.h"
#include "MilkProducer.h"
#include <string>
```

Classes

- class **Horse**

5.21 HorseMilk.h File Reference

```
#include "FarmProduct.h"
```

Classes

- class **HorseMilk**

5.22 Land.h File Reference

```
#include "Cell.h"
```

Classes

- class **Land**

5.23 LinkedList.h File Reference

```
#include <initializer_list>
```

Classes

- class **LinkedListNode**< **T** >
- class **LinkedList**< **T** >
- class **LinkedListNode**< **T** >

5.24 LinkedListDriver.cpp File Reference

```
#include <bits/stdc++.h>
```

Classes

- class **LinkedList**< **T** >
- class **LinkedListNode**< **T** >
- class **LinkedList**< **T** >

Functions

- template<class T >
  ostream & **operator**<< (ostream &os, **LinkedList**< T > l)
- int **main** ()

5.24.1 Function Documentation

5.24.1.1 main()

```
int main ( )
```

5.24.1.2 operator<<()

```
template<class T >
ostream& operator<< (
             ostream & os,
              LinkedList< T > l )
```

5.25 LivingThing.h File Reference

```
#include "Point.h"
#include "Cell.h"
#include "Direction.h"
```

Classes

- class **LivingThing**

5.26 MeatProducer.h File Reference

```
#include "../Point.h"
#include "../Cell/Cell.h"
#include "FarmAnimal.h"
```

Classes

- class **MeatProducer**

5.27 MilkProducer.h File Reference

```
#include "../Point.h"
#include "../Cell/Cell.h"
#include "FarmAnimal.h"
```

Classes

- class **MilkProducer**

5.28 Mixer.h File Reference

```
#include "Facility.h"
```

Classes

- class **Mixer**

5.29 Ostrich.h File Reference

```
#include "../Point.h"
#include "../Cell/Cell.h"
#include "EggProducer.h"
#include <string>
```

Classes

- class **Ostrich**

5.30 OstrichEgg.h File Reference

```
#include "FarmProduct.h"
```

Classes

- class **OstrichEgg**

5.31 Player.h File Reference

```
#include "LivingThing.h"
#include "LinkedList.h"
#include "FarmAnimal.h"
#include "Cell/Cell.h"
#include "Product/BeefChickenOmelette.h"
#include "Product/BeefMuttonSate.h"
#include "Product/SuperSecretSpecialProduct.h"
#include "Point.h"
```

Classes

- class **Player**

5.32 Point.h File Reference

Classes

- struct **Point**

5.33 PointDriver.cpp File Reference

```
#include "Point.h"
#include <bits/stdc++.h>
```

Functions

- int **main** ()

5.33.1 Function Documentation

5.33.1.1 main()

```
int main ( )
```

5.34 Product.h File Reference

Classes

- class **Product**

Macros

- #define **Category** unsigned int
- #define **CHICKENEGG** 6
- #define **CHICKENMEAT** 7
- #define **BEEFCHICKENOMELETTE** 8
- #define **BEEFMUTONSATE** 9
- #define **COWMILK** 10
- #define **COWMEAT** 11
- #define **DUCKMEAT** 12
- #define **HORSEMILK** 13
- #define **OSTRICHEGG** 14
- #define **SHEEPMEAT** 15
- #define **SUPERSECRETSPECIALPRODUCT** 16

5.34.1 Macro Definition Documentation

5.34.1.1 BEEFCHICKENOMELETTE

```
#define BEEFCHICKENOMELETTE 8
```

5.34.1.2 BEEFMUTONSATE

```
#define BEEFMUTONSATE 9
```

5.34.1.3 Category

```
#define Category unsigned int
```

5.34.1.4 CHICKENEGG

```
#define CHICKENEGG 6
```

5.34.1.5 CHICKENMEAT

```
#define CHICKENMEAT 7
```

5.34.1.6 COWMEAT

```
#define COWMEAT 11
```

5.34.1.7 COWMILK

```
#define COWMILK 10
```

5.34.1.8 DUCKMEAT

```
#define DUCKMEAT 12
```

5.34.1.9 HORSEMILK

```
#define HORSEMILK 13
```

5.34.1.10 OSTRICHEGG

```
#define OSTRICHEGG 14
```

5.34.1.11 SHEEPMEAT

```
#define SHEEPMEAT 15
```

5.34.1.12 SUPERSECRETSPECIALPRODUCT

```
#define SUPERSECRETSPECIALPRODUCT 16
```

5.35 Sheep.h File Reference

```
#include "../Point.h"
#include "../Cell/Cell.h"
#include "MeatProducer.h"
#include <string>
```

Classes

- class **Sheep**

5.36 SheepMeat.h File Reference

```
#include "FarmProduct.h"
```

Classes

- class **SheepMeat**

5.37 SideProduct.h File Reference

```
#include "Product.h"
```

Classes

- class **SideProduct**

5.38 SuperSecretSpecialProduct.h File Reference

```
#include "../LinkedList.h"
#include "SideProduct.h"
```

Classes

- class **SuperSecretSpecialProduct**

5.39 Truck.h File Reference

```
#include "Facility.h"
```

Classes

- class **Truck**

5.40 Well.h File Reference

```
#include "Facility.h"
```

Classes

- class **Well**

5.41 World.h File Reference

```
#include "Player.h"
#include "LinkedList.h"
#include "Cell/Cell.h"
#include "FarmAnimal/FarmAnimal.h"
```

Classes

- class **World**

Index